Universitatea POLITEHNICA din București
Facultatea de Automatică și Calculatoare
Departamentul Calculatoare

Lucrare de Diplomă

Punct de acces mobil aerian pentru retele de senzori folosind drone

Autor

Ioan Deaconu

Coordonator: As. Drd. Ing. Andrei Voinescu

București, Septembrie 2014

University POLITEHNICA of Bucharest
Faculty of Automatic Control and Computers
Computer Science and Engineering Department

Diploma Thesis

Aerial Mobile Gateway for Wireless Sensor Networks utilizing drones

Author

Ioan Deaconu

Supervisor: As. Drd. Ing. Andrei Voinescu

Bucharest, September 2014

Contents

Abstract

This thesis proposes a new way of implementing a mobile gateway for Wireless Sensor Networks that simplifies applications running in remote locations, where maintenance is difficult. Wireless Sensor Networks islands need a gateway connection in order to reach the outside world, but this is difficult to provide in all instances. The solution to this problem is to use a gateway mounted on a UAV that can reach those islands and extract data from them. This solution has been proven to be successful but adoption is low because of the high cost and the technical background needed to operate it. We propose a simpler, easier to operate and cheaper solution for this problem.

Keywords Wireless Sensor Networks, drones, sparrow, gps

Acknowledgements

Chapter 1

Introduction

Technology has been present in our lives for a long time and we have become so accustomed to it that we do not even realize when we are using it. This technological advance has lead to the development of small devices with communication capabilities and low power consumption. These devices can form a wireless sensor network which can be used to collect data from the environment and send it to a gateway for further processing.

The applications of a Wireless Sensor Network are unlimited in number. For example they can be used to monitor crops, to detect possible forest fires, to detect the presence of animals or vehicles in certain areas, to track and monitor doctors in hospitals, to guide interactive toys, to detect and monitor car thefts etc. In the last ten years, integration of wireless sensor networks with unmanned aerial vehicles had been tested and proven to be successful. However, previous implementations, described in chapter 2, are complicated, difficult to operate and too expensive for the general public.

One goal of a Wireless Sensor Network is to collect data and send it to a device called a gateway. A frequent scenario is that of a number of nodes deployed to monitor an area that is hard to reach for various reasons. Installation of a gateway in such an area is almost always impossible, but usually the area is accessible to an unmanned aerial vehicle. The way the UAV reaches that area can be either by being remote-controlled by a human operator or by following a list of predetermined waypoints specified by the user. When the UAV reaches the nodes, it can start retrieving and saving the data so it can be sent back home. The data collected by the drone should be accessible at any time, if the UAV is powered.

The solution that we propose is based on a very popular and easy to use drone, the AR Parrot Drone 2.0, and the Sparrow Family [VTD13] of sensor nodes, developed at University POLITEHNICA of Bucharest by Andrei Voinescu.

Chapter 2

Related Work

The related work is starting the expand and new researches propose new ideas, but generally speaking they tend to focus on ways of collecting the data from the nodes. The objective in their articles is to see if an UAV can be integrated with a wireless sensor networks. The conclusion is generally positive, but a big problem in adopting their research in real life scenarios is represented by the high costs of the equipment they used and the necessary knowledge required to setup and operate the equipment.

The general experiments presented in the UAV and WSN integration research are the following:

- Using nodes signal to perform course corrections for dynamic navigation
- Data muling from nodes
- Using drones to deploy a new node in order to expand or to fix a problem in the network
- Using drones to determine ground military activity [ASSC02]

2.1 Standard WSN Protocols

The protocols implemented in Wireless Sensor Network are based on surrounding node discovery in order to build a topology and find the best way they can multi hop data to the gateway. This approach works best in a static environment, but in a dynamic environment or an eviroment were the distance between nodes is too big or the time between two data packets is too big, the network convergence will be slow or not even possible.

2.2 UAV experiments with Wireless Sensor Networks

[TMCH08]

The experiment consisted of using ground nodes that had a GPS position assigned. The UAV plane would performed course correction after receiving the current GPS position from the node in order to calculate the best path for muling the data from the nodes.

The advantage of using a plane used for the experiment is the longer range and higher speed that it can offer against a quad copter or a similar design. But the high speed creates the problem of maneuverability. The plane has a turning range of 400 meters while the drone can almost turn on the same spot.

2.3 Crop Monitoring

[VSB+11]

A research of using a drone for crop monitoring has been conducted at a vineyard. Their system was comprised of a unmanned quad copter, an Arduino board with a GPRS module for long distance communication with the drone and ZigBee and Crossbow's TelosB as wireless sensing nodes. The drone was not controlled via the long-distance link, but through a Spektrum DX7SE 2.4 GHz remote control.

They demonstrated that a preprogrammed UAV can be used to monitor multiple crops where a standard WSN could not be deployed because of the unique constrains imposed by the environment.

The cost of the implementation was relatively high compared to ours, the remote is 300$, the same as the entire drone that we propose and the TelosB is 99$. This data suggests that for their experiment the drone, communication module and the remote control were half the cost of the equipment.

Another problem was that they were not saving the data locally, but sending it back to the base station where it was processed and saved. This can represent a problem because the system cannot function properly unless a base station is supplied.

2.4 Aware platform

[OBLC+07]

The Aware platform, proposed by Ays. Egül Tüysüz Erman, Lodewijk Van Hoesel and Paul Havinga from University of Twente, is a platform that integrates WSNs, UAVs, and actuators into a disaster response setting and provides facilities for event detection, autonomous network repair by UAVs, and quick response by integrated operational forces.

They use multiple UAVs to deploy new nodes that will replace the damaged ones and check if they function. The entire system still relies on a sink to collect the data and to send them to a base station.[EHHW08]

Chapter 3

Hardware Platform

In this chapter we will present the hardware platforms used.

Because we wanted to emphasize not only a new way of acquiring data, but also a simple and low cost one, we have selected The Parrot AR.Drone 2.0 as the work horse that will carry the mobile gateway which communicates with the nodes from our Sparrow Family. The drone provides several key features that we require, such as a Linux embedded system, a mobile platform with the possibility of autonomous flying and sufficient flight time for our needs.

In order to keep the price low and a small footprint, the Sparrow Family has a surface mounted antenna on the PCB. The power of the antenna is 2 dBi and it is perfect for applications were size is a constraint. When the size can be overlooked, an external antenna can be mounted to extend the range of the device.

The cost of our solution depends on the configuration and size of the WSN islands. The cost is 300$ for the drone, 35$ for a node and 8 $ for a high gain antenna.

Example cost for two different types of networks:

- An area that needs a high density of nodes, but covers a small surface, does not require to nodes to have the external antenna. The cost would amount to 700$
- An area that needs a low density of nodes, but covers a larger surface, or are place in a high vegetation or interference area will require the nodes to be equipped with an external antenna. The cost would amount to 800$

3.1 The Parrot AR.Drone 2.0

[Par12]

Parrot AR.Drone is a Wi-Fi radio controlled flying quad copter built by the French company Parrot. The original drone was released in 2010 and in 2012 it was replaced by version 2.0. Since the launch of the original AR.Drone, more the half a million units have been sold, making it one of the, if not, the most popular drone on the market.[BRA13]

The reason of its success is not entirely due to the relatively low price of around 300$ but also because it is very easy to learn how to control the drone and also because of the USB port that accommodate any device using that interface and the Linux operating system, making it incredibly versatile and very easy to integrate in our system.

Figure 3.1: *The Parrot AR.Drone 2.0*[par]

Because of those reasons, the Drone has a number of after-market modules that can be attached to it, such as the Flight Recorder GPS Module. This module has a built-in storage of 4GB for video recording purposes and a built in GPS receiver. This allows the drone to follow a predetermined path of way points and to return back from where it took off automatically, all within the limit of the Wi-Fi connection with the control device.

In order to properly accommodate the Sparrow Dongle, the hull had to be carved. Also the required external antenna of the dongle was mounted on top of the polyester cover and a small counterweight has been glued at the opposite side in which the dongle is located. The counterweight acts as a ballast that keeps the drone level.

The Parrot AR.Drone 2.0 specifications are :

- 1GHz 32 bit ARM Cortex A8 processor with 800MHz video DSP TMS320DMC64x

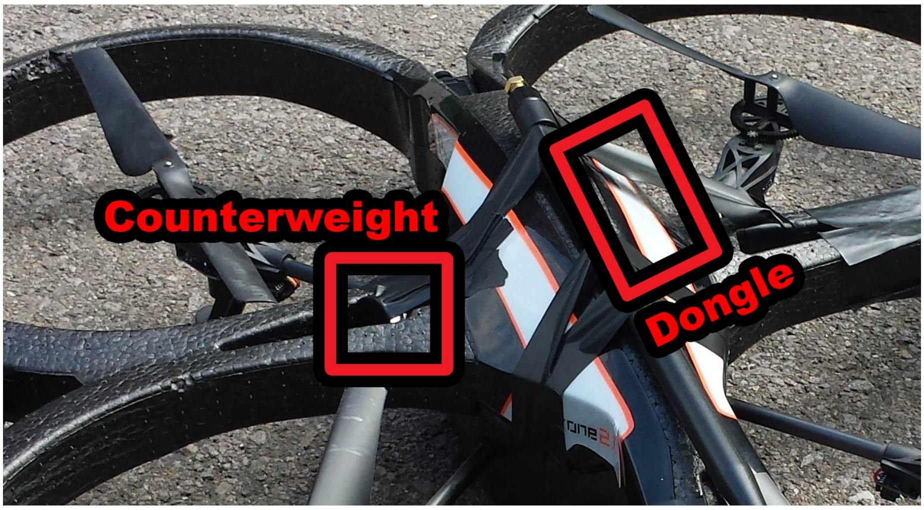


Figure 3.2: *The counter weight need to balance the drone*

- Linux 2.6.32
- 1Gbit DDR2 RAM at 200MHz
- USB 2.0 high speed for extensions
- Wi-Fi b,g,n
- 3 axis gyroscope 2000°/second precision
- 3 axis accelerometer +-50mg precision
- 3 axis magnetometer 6° precision
- Pressure sensor +/- 10 Pa precision
- Ultrasound sensors for ground altitude measurement
- 60 fps vertical QVGA camera for ground speed measurement
- 30 fps 720p front mounted camera

3.2 The Sparrow Family

The Sparrow Family, composed of the Sparrow Dongle and SparrowV3.2, use a 2,4 GHz wireless network as a medium of communication.

The main component of this family is the ATMega128RFA1. It is an 8bit micro-controller from Atmel that has an on-chip 2.4 GHz wireless transceiver. On-chip transceivers occupy no extra space and require little extra electronics to

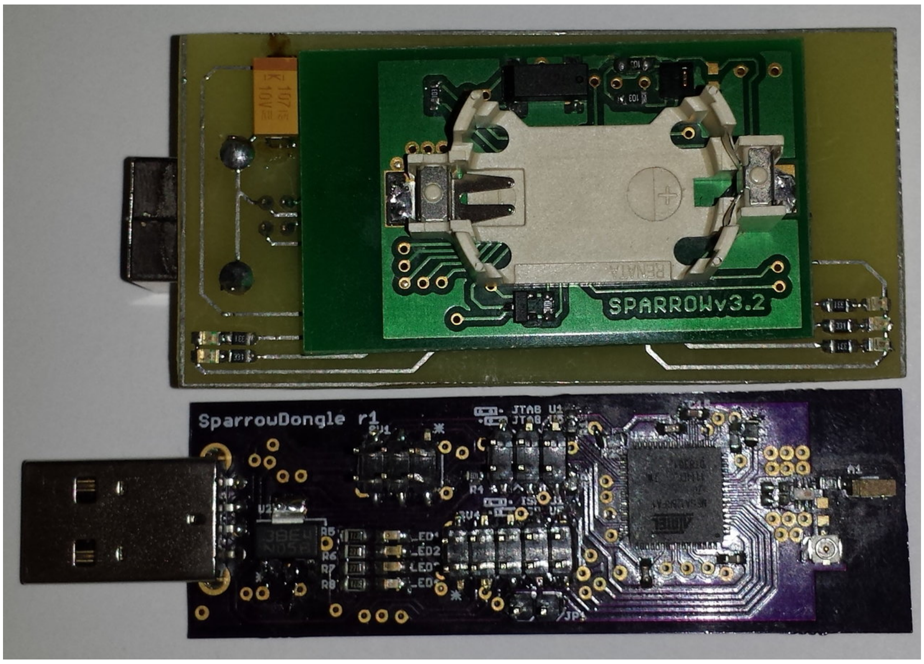


Figure 3.3: *The Sparrow Dongle next to the SparrowV3.2*

operate, making the resulting boards very small. The on-chip transceiver allows more energy-efficient operating modes, and facilitate higher bandwidth transfers between the micro-controller's main memory and the transceiver frame-buffer, all of which are important improvements in the field of wireless sensor networks.

The signal received or sent by the wireless transceiver can be boosted by attaching an external antenna. For example, in an ideal situation, with no interferences from the outside world , an 8 dBi omni-directional antenna mounted on both communication devices would amount to an around 200 meters of communication range, well over the 70 meters measured with the default antennas.

This distance is achieved with the RX and TX at full power. A higher battery life can be achieved by reducing RX and TX power, but the maximum communication range will be shortened. The reduction can be compensated by installing a high gain antenna, but this would significantly increase the cost .

3.2.1 The Sparrow Dongle

The link between the the wireless sensor networks and the rest of the digital world, the Sparrow Dongle is the gateway of the Sparrow Family. The Dongle can be connected to any device that has an USB port and can support USB CDC with ACM module (USB Communications Device Class with Abstract Control Mode).

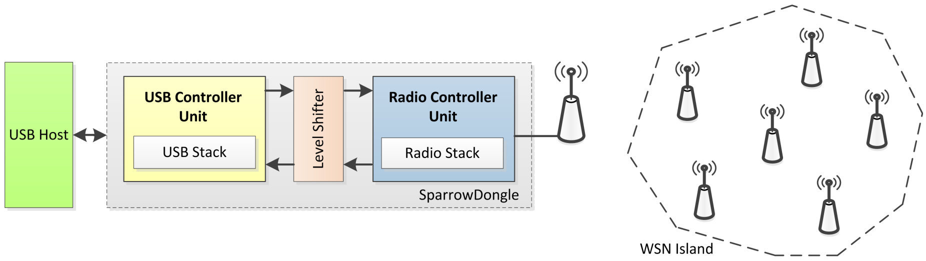


Figure 3.4: *SparrowDongle stick architecture*[VTD13]

The dongle uses Atmega32U4 as a dedicated USB Controller unit. This design allows the RF controller to run any RF communication stack without having the USB code intrude on key timings.[VTD13]

3.2.2 The SparrowV3.2

The other member of The Sparrow Family, the SparrowV32 nodes are responsible of collecting data from the surrounding environment and sending it to the Sparrow Dongle. The collected data depends on the sensors attached to the SparrowV3.2. The standard implementation has a light, temperature and humidity sensor. Besides this information, it also sends the battery level.

The SparrowV3.2 can be modified by adding additional sensors for a better understanding of the environment.

The node can be powered by a non-rechargeable battery, a rechargeable one or by a super-capacitor. Even though the super-capacitor can contain only a small charge for a small amount of time, the stored energy is sufficient enough to keep the the node up and running for at least a day.

The advantages of the super-capacitor over the rechargeable battery are the following:

- it can charge and discharge almost instantaneously
- it has a very high number of charge/discharge cycles that allows the node for a
- it does not present over years the same symptoms as an aging battery
- it is far less pollutant than a standard battery
- it will allow a higher maintenance free time than a battery

The rechargeable battery and the capacity can be recharged from a solar panel, a wind turbine, or any other available renewable sources of energy.

Chapter 4

Software Implementation

The solution is divided into different modules that run independently but communicate with each other to achieve the main goal. The separation of software modules allows for future features to be added easily.

The main modules are installed in the AR Parrot Drone and the Android FreeFlight 2.0 application.

The Sparrow Dongle gateway is always in a listen-for-data state and dumps any data received on the serial. When it receives the data, it sends back an ACK message back to let the SparrowV3.2 node to know that it can begin sending the entire stored data to the mobile gateway.

The SparrowV3.2 node is sending periodically a small data packet to check if a gateway is available. It stores the data accumulated over the period when no reply is given to it. When it receives the ACK message from a mobile gateway it starts sending the stored data to the gateway. The data sent can vary, from sensor readings to debugging information in order to check the state of the Wireless Sensor Network.

The data gathered by the gateway is saved into different files in the AR Parrot Drone's internal memory. The files also contains information such as the node identification tag and time of the transfer. The data can be accessed at any time by any device connected to the drone's wireless network port 4242 via FTP.

The drone also processes some of the collected data to provide real time HUD information, such as signal strength, last connection time and number of discovered nodes. This information is sent to the controlling device through a socket connection.

The controlling device, PC or android, will gather the information and display them to the user.

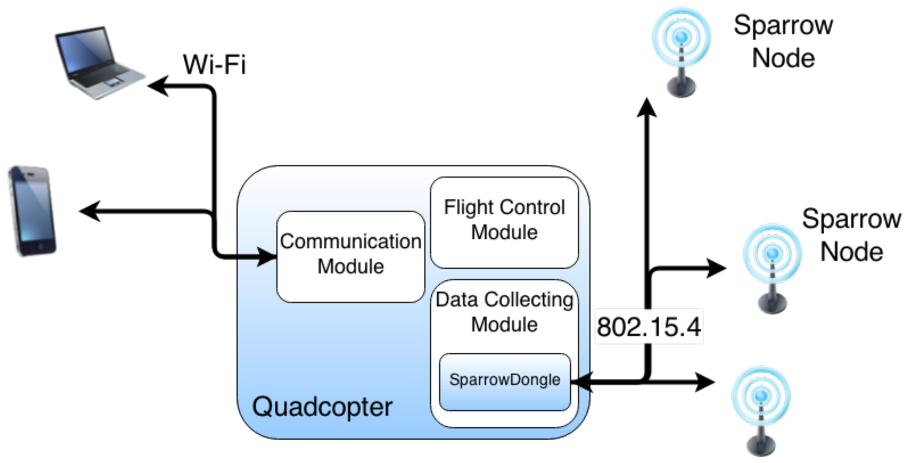


Figure 4.1: *Modules and connections between them and devices*

4.1 The

The default software with which the drone is delivered does not recognize our USB dongle. This behavior was expected because the drone has a stripped down version of Linux. In order for the drone to recognize the dongle, a module had to be compiled for the specific CPU architecture and operating system that were installed on the drone.

The required module is the module cdc_acm[cdc10]. This module is responsible for emulating serial ports over USB. The module creates a file named /dev/tty-ACM0 that is linked with the dongle. The file can be read just like any other file .

4.2 The Debug Module

When performing modification to the existing code, a debug enabling option would speed up the process. The module would allow for displaying control message to the user console even when the process is running in background.If the messages would be activated all the time, they would slow down the execution speed of the process.

In order to see those messages, the debug option must be activated and then this simple command will show them:

Listing 4.1: Simple display message command

```
p=$(pidof read) && strace -p $p
```

Enabling the debug is just a matter of setting a define from 0 to 1, recompile and upload the code to the drone to see the messages.

Listing 4.2: Debug and timing defines

```
/* activates/deactivates printf debug information*/
#define DEBUG_ON 0
/* delay yime in microseconds*/
#define DELAY_US 100000
#define DEBUG_PRINT(a...) { if(DEBUG_ON) printf(a); }
```

In certain parts of the modules a sleep action is needed in order to wait for an action to be executed. The value can be changed to any level, but you must be careful in doing this. A small delay will send data more often but it could use allot of processing power, while a big delay could be to slow for the data to be usable. Now the delay is set at 100 ms, for a 10 times per second data update.

4.3 The Data Collecting Module

The module saves the collected data into the drone's internal memory and pases the data on to the communication module, which will display on the controller interface certain data: number of nodes currently connected to the Dongle, the signal strength, if the Dongle is connected etc.

This module, besides the main purpose and similar to the other modules, has some extra features that are design to make the solution more user friendly and easier to improve in the feature.

4.3.1 Modules intercommunication

The memory area in which the information sent to the user is saved is shared between this module and the communication module. The interaction method between these two modules belongs to the consumer-producer archetype, where the Data Collecting Module can be associated with the producer side and the Communication Module with the consumer side.

A sensible issue with this approach regards possible deadlocks. This is prevented with the use of a mutex construct that allows only one thread at a time to modify the data.

Listing 4.3: Data Collection use of mutex

```
pthread_mutex_lock(&data_lock);
add_node_data(get_current_timestamp(),read_data + 7);
pthread_mutex_unlock(&data_lock);
```

The mutex is used similarly in the Communication Module when it consumes the information.

4.3.2 Fault tolerance

Because the Dongle is connected to an USB port on a machine that has a lot of vibrations, it might disconnect / reconnect for a very short period of time, so this module has been designed with multiple USB disconnects and reconnects without the need to reset the Drone. This information is vital, because you can check if the Dongle is still connected to the drone without the need to inspect it visually or to connect to a debug terminal.

Besides the possible dongle USB disconnects, an out of range signal loss may be experienced. It this happens, the drone will hover until the connection is reestablished.

4.4 The Communication Module

All of the information gathered by the Data Collecting Module would be useless if it cannot be accessed easily.

This module, as the name suggests, handles the communication of this this crucial information back to the user.

Being a different module, with different attributions than the Data Collecting Module, it has an entire Linux process dedicated to it for 3 important reasons:

1. The approach of having a process per module allows the modules to run independently of each other;
2. The Data Collecting Module can collect the data from the Dongle as soon as this is available;
3. If the Communication Module stops working, the Data Collecting Module can keep collecting data, so complete failure of the system is avoided;
4. System processes can be restarted in case of failure.

4.4.1 Socket with connection reset

The communication is done through socket connections listening on port 8888. The server running on the drone accepts only one connection at a time.

If a connected client decides to disconnect before or while a write operation is in progress, a SIGPIPE error signal will be thrown, stopping all the modules. This is prevented by ignoring the signal, forcing the write action to return a EPIPE, and exiting gracefully.

The main process will use the callback accept_socket_connection to reestablish a new connection. Once a connection is established, it will send information once every DELAY_US microseconds. The program was configured and tested with a 100 ms wait period that leads to a ten times per second information update.

This delay is required because:

- If data is send too often, the socket might be flooded and stop sending the data.
- If there was no delay, it would occupy too much processor time both for the drone and the controlling device.

4.4.2 JSON Encoding of Data

[jso]

JSON is an open-standard that uses text to encode data and it is an alternative to XML. Derived from the JavaScript scripting language, it is a language-independent data format available in most of the programing languages.

JSON is best suitable for this application as a data encode format because it is data oriented, unlike XML which is document oriented. Also it is very easy to encode because it has a code like structure, the result is smaller than the XML alternative and all devices can decode it.

The informations encoded by the drone are:

- Dongle connection status
- An array containing node data
  - Node unique ID
  - Last connection time of the node to Dongle
  - The power of the received signal

4.5 SparrowV3.2 module

The Sparrow sends a small packet a fixed interval. If the packet is received by the Dongle, it will sent a specific ack just to the sparrow that sent the packet. When the node receives the ack, it will try to send all the stored data to the drone, from the oldes one to the newest one. The data is stored in a cicular linked list that has a fixed zise. If the list is full, the oldest data will be replaced by a new data.

The drone will save the receied data, localy in files that have the name composed from the timestamp of the current sesion and the unique id of the node.

4.6 Android application modules

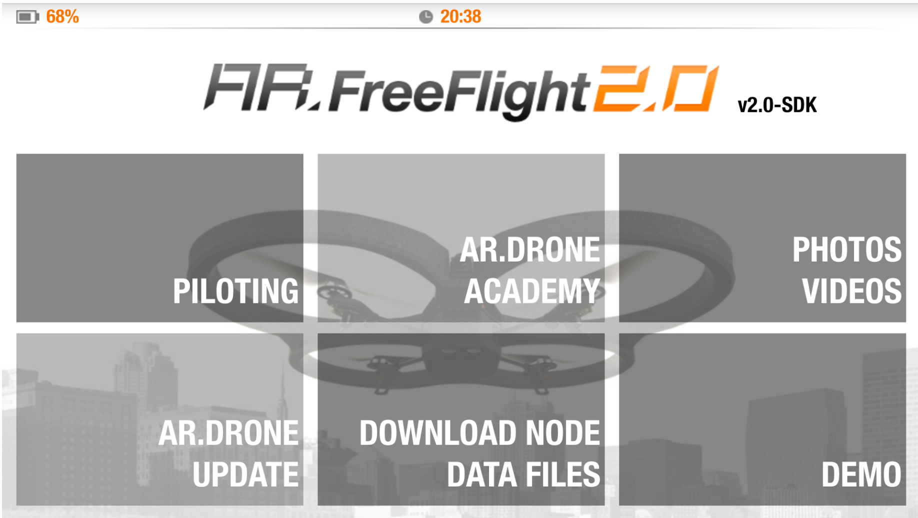


Figure 4.2: *ARFreeflight modified application*

Being an open-source platform we have modified the AR Freeflight 2.0 Android application to communication with our new modules added to the drone.

Android fairly imposes the use of the background process class AsyncTask when you have to use communication protocols like http, ftp, sockets because this prevents the UI process from being stuck in communication and not responding to user actions.

The class offers 5 very important methods that can be overwritten, 3 running on the main UI process, that prepare data before and after execution, publish the progress or simply cancel at any step, and 1 running on the actual background process.

4.6.1 Display information module

The Piloting screen of the application has been modified to the display the received data from the drone.

The information displayed is comprised of the state of the dongle being functional or not. Besides that, it can show the signal strength, unique id and last connection time of up to 9 nodes sorted descending after their signal strength.

4.6.2 FTP communication module

The drone has a built-in FTP server that can be configured to allow access to any folders/files on the drone. We have configured the drone so that the folder

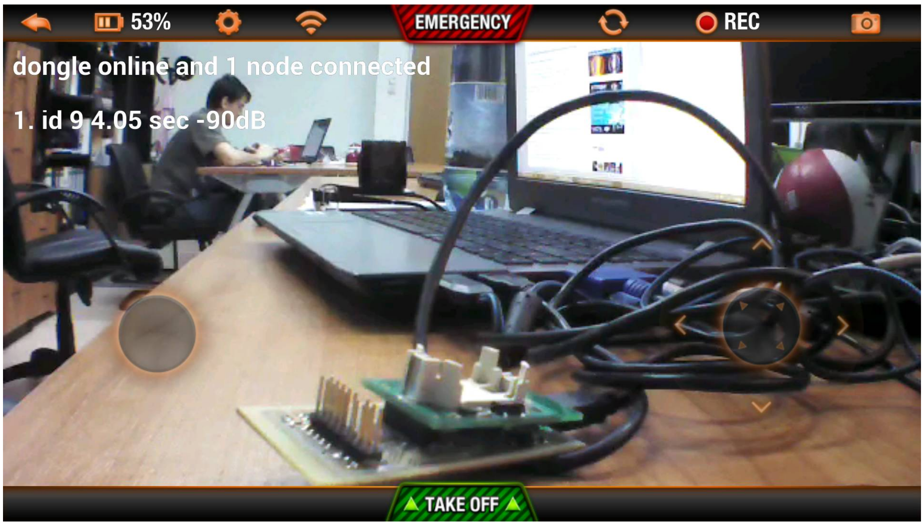
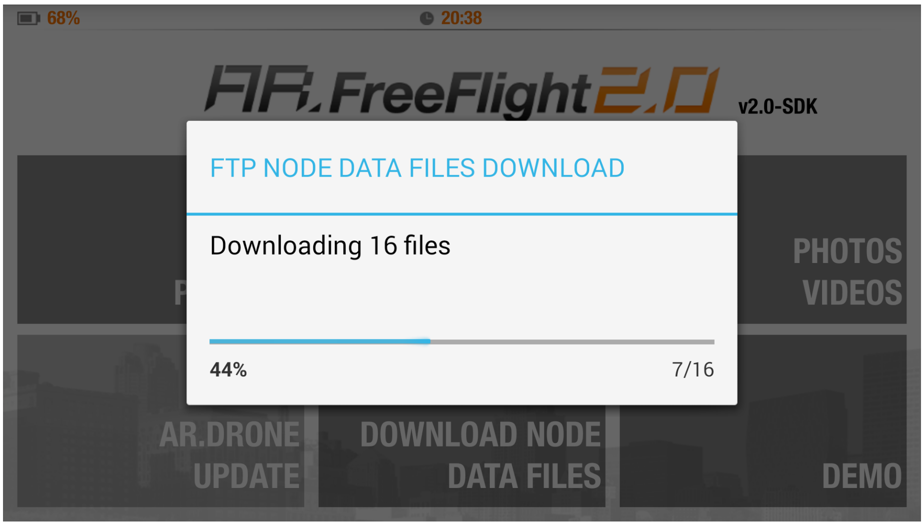


Figure 4.3: *ARFreeflight modified Piloting Screen*



W

Figure 4.4: *ARFreeflight FTP downloading files*

in which the data is saved can be accessed at any time using the 4242 port by any device that has FTP client capabilities.

This feature is add to the Android application as well to have a better out of the box experience.

The application will download all the files from the drone to the local storage of the user's Android device while displaying the progress.

Chapter 5

Testing

The tests that we have conducted highlights the strengths of the platform but also reveals some of its weaknesses. The most important characteristics of this experiment the maximum hight that the drone can achieve with extra gear, the maximum range of the wireless network and the maximum range of the dongle .

Tests show the maximum range at which a drone can still communicate with base or with a SparrowV3.2,

5.1 Scenario

The testing method were conducted using a dongle with the standard antenna, a sparrowV3.2, a 3 dBi and a 8 dBi external antenna. The nodes were placed on the ground with the antenna directed upwards.

Because both the drone and nodes use 2.4 GHz network, the drone was configured to use channel 11 and the nodes to use channel 1 to prevent signal interference.

Signal testing was performed by walking with the done in one hand and the phone in the other until the drone could not receive packets from the nodes. This test was not done by flying the drone because the exact point at which the signal is lost can not be controlled with the desired precision.

5.2 Results

5.2.1 Signal range

The top mounted antenna worked, but for a better signal, the antenna should always be positioned on the bottom of the drone because in this way, the antenna will have a clear path to send and receive signal and not have the signal blocked

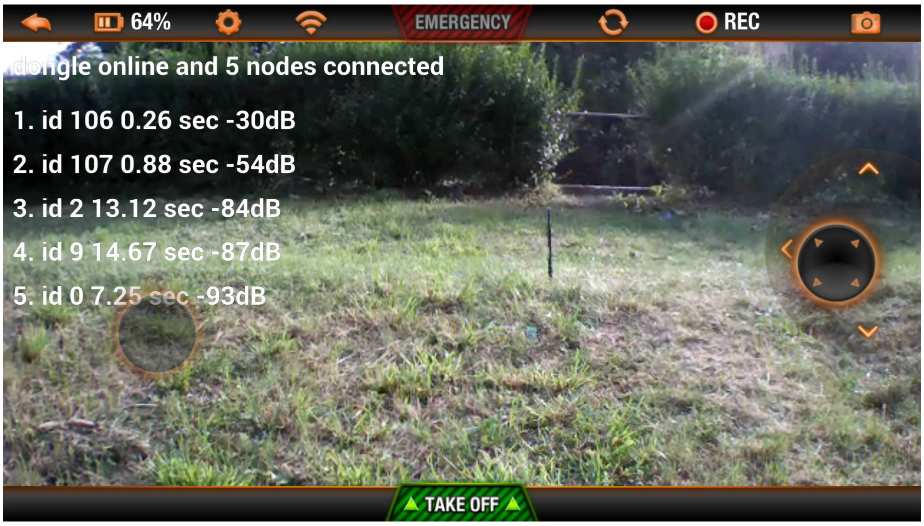


Figure 5.1: *Parrot discovering new nodes before takeoff*

Figure 5.2: *Measured signal distance*

by the entire drone and its avionics. The problem with this drone is that on the bottom of it, there are sensors that help her determine the ground speed, ground distance and the antenna could obstruct some of the sensors.

The range test results proved that the drone could receive signal from the nodes at maximum distance of 325 meters and a clear line of sight. If the node is obstructed by an object the maximum distance will decrease. For example, a node that is is placed behind an air conditioning unit and a 2 dBi antenna will have a range of just 70 meters.

5.2.2 Drone stability

Even though the antenna was mounted on top of the parrot, because it is a high gain antenna the signal received is very strong.

The antenna extended the signal range but also added weight. The dongle is mounted on the side of the drone due to the position of the USB. Because the dongle is not centered on the drone, a counterweight had to be glued on the opposite side on the outer shell to maintain the balance of the drone.

The drone was relatively stable during the test, but a better stability and flight control could be obtained if the dongle will be shrunk and a lighter antenna is mounted.

Figure 5.3: *Top mounted antenna for better signal*

5.2.3 Maximum height and maneuverability

The total added weight is 75 grams. Even though it does not sound that much, it does have a substantial effect on the drone. The maximum height it can reach is 171784 versus the 3235 it can reach without the added weight. The maneuverability is also affected, the drone response not being as sharp as before, but it is still good.

5.2.4 Problems

The kernel module needed for the the dongle does not recognize the dongle if it is plugged in the drone when it is powered up. The fix is to power up the drone and then plug in the dongle, but this is more difficult then it might appear because every time this action is performed the hull must be repositioned.

Chapter 6

Conclusions

With the current technological advances, we are more and more used to things that make our lives more simpler and give us informations at the tip of our finger.

The Wireless Sensor Networks are expanding more and more because they make our lives simpler by giving us informations about our surroundings. But the standard way of creating this islands is not always feasible.

The main purpose of this thesis was to bring an alternative at how the data is muled from the Wireless Sensor Network islands. The solution, a truly mobile gateway is a viable one and it can be implemented with a relatively low cost and no extra knowledge, other than knowing how to use a smartphone.

The solution is not in its final state and can be improved by adding new features.

A feature that can be used in conventional wireless sensor networks is to determine the source of communication failure. If the gateway detects that the network has a communication problem and not all of the previous nodes can be reached, this information will be send to the drone and it will try to find and determine which nodes are working properly and which nodes are not.

The AR Parrot Drone 2.0 can perform autonomous flight with a GPS module, but only while it is still in the range of the Wi-Fi connection. This feature will allow the drone to fly without the need to still be connected through Wi-Fi to a controlling device. Also, a different autonomous flight can be performed without a GPS module if the signal strength of the nodes is used to performed flight correction and determine the speed and direction of the drone.

Besides the practicality aspect of the solution, it can be used as a fun instrument to.

A treasure hunt game can be played by placing the nodes with information regarding the location of the treasure at certain hidden spots and trying to find the treasure by following the clues provided by the drone.

Other fun game that can be played is hiding the nodes in a certain zone. The players must find as many nodes as possible in that zone before they run out of juice. The winner will be the player that found the biggest number of nodes and has the best time.

Bibliography

[ASSC02] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002. [cited at p. 4]

[BRA13] ALEX BRACETTI. The 10 Best Drones You Can Buy Right Now. http://www.complex.com/pop-culture/2013/03/10-cool-drones-you-can-buy-right-now/parrot-ardrone-20, March 2013. [cited at p. 8]

[cdc10] USB ACM. http://wiki.openmoko.org/wiki/CDC_ACM, March 2010. [cited at p. 13]

[EHHW08] Aysegül Tüysüz Erman, LV Hoesel, Paul Havinga, and Jian Wu. Enabling mobility in heterogeneous wireless sensor networks cooperating with uavs for mission-critical management. *Wireless Communications, IEEE*, 15(6):38–46, 2008. [cited at p. 6]

[jso] Json Standard. http://www.json.org/xml.html. [cited at p. 16]

[OBLC+07] Anibal Ollero, Markus Bernard, Marco La Civita, Lodewijk van Hoesel, Pedro J Marron, Jason Lepley, and Eduardo de Andres. Aware: Platform for autonomous self-deploying and operation of wireless sensor-actuator networks cooperating with unmanned aerial vehicles. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–6. IEEE, 2007. [cited at p. 5]

[par] Parrot Drone. http://img.clubic.com/04862120-photo-parrot-ar-drone-2-0.jpg. [cited at p. 8, 26]

[Par12] AR Parrot. Drone. *Available: ardrone. parrot. com*, 75, 2012. [cited at p. 7]

[TMCH08] Steven K Teh, Luis Mejias, Peter Corke, and Wen Hu. Experiments in integrating autonomous uninhabited aerial vehicles (uavs) and wireless sensor networks. 2008. [cited at p. 5]

[VSB+11] João Valente, David Sanz, Antonio Barrientos, Jaime del Cerro, Ángela Ribeiro, and Claudio Rossi. An air-ground wireless sensor network for crop monitoring. *Sensors*, 11(6):6088–6108, 2011. [cited at p. 5]

[VTD13] Andrei Voinescu, Dan Tudose, and Dan Dragomir. A lightweight, versatile gateway platform for wireless sensor networks. In *Networking in Education and Research, 2013 RoEduNet International Conference 12th Edition*, pages 1–4. IEEE, 2013. [cited at p. 3, 11, 26]

List of Figures

List of Tables

Listings